PointDCCNet: 3D Object Categorization Network using Point Cloud Decomposition

Siddharth Katageri, Sameer Kulmi, Ramesh Ashok Tabib, Uma Mudenagudi,
Center of Excellence in Vision Intelligence (CEVI), KLE Technology University,
Hubballi, Karnataka, INDIA
siddharth.katageri19@gmail.com, sameerkulmi04@gmail.com, ramesh_t@kletech.ac.in, uma@kletech.ac.in

Abstract

In this paper, we propose Point Decomposition Network (PointDCCNet) for 3D object categorization using point cloud decomposition. In the recent technologies for 3D data capture, point clouds have a surge in demand due to their simpler representation and computations. The point cloud analysis requires robust methods for feature extraction to tackle the permutation invariance and unorderdness in point sets and finds application in categorization, refinement, and super-resolution of 3D data. We propose a novel PointDCCNet towards the decomposition of point clouds into primitive geometric shapes, namely plane, sphere, cone and cylinder; and use it as a clue towards modelling a classifier for 3D object categorization. The decomposition of point clouds provides a geometrical signature of the 3D object towards categorization. We show the decomposition of 3D data into primitive shapes which assists the model in the categorization of 3D objects. We demonstrate the results using benchmark datasets and compare them with state-of-the-art techniques.

1. Introduction

Point clouds are a set of data points in 3D representing the shape in the simplest form. Recently, many 3D point cloud applications like robotics [32], autonomous driving [26], architectural and archaeological heritage sites digitalization [23] have seen rising demands due to its easy computations and storage. These require accurate and robust 3D point cloud data for processing and analysis. Photogrammetry, structured light, ToF sensors and LiDAR sensing technologies produce an unordered set of 3D points trying to capture the object surface and also there are huge man-made 3D data repositories which widens the opportunities of working on 3D data processing. However, the data obtained by sensors and such repositories are represented as digitailized point clouds or meshes. This simplest

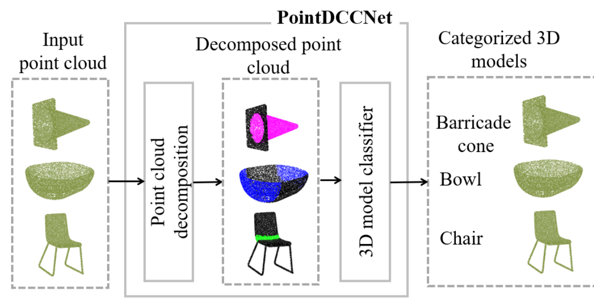

Figure 1. PointDCCNet for 3D object categorization

level of representation makes it potentially challenging to geometrically understand and operate on 3D data, due to missing structural information. To address this issue, in this work we look at this as a problem of 3D object decomposition, which maps each point to a geometric primitive shape, defining its unique set of geometric signature thus, acquiring structural information. Knowing the underlying geometry will greatly assist many 3D analysis/generation tasks. We show our framework for 3D object categorization in Figure 1, which takes into account the underlying geometry of the 3D object for categorization.

Previous works in computer vision for primitive shape detection and shape decomposition were performed using classical methods like RANSAC [5] and its other variants [42, 24, 3, 16]. A RANSAC-based framework, [34] tries to solve this issue by considering it as a problem of primitive shape fitting and is able to detect different primitive types on a dense point cloud. [21] improves on this by optimizing on extracted primitives based on relation between the primitives. These methods are effective to outliers. However, weakness of RANSAC based approaches are that the manual parameter tuning is labour-intensive. This demands for careful supervision and makes it non-scalable for large datasets [15].

Another direction of work focuses towards incorporating the achievements of convolution neural networks on regular grid data (2D data) analysis [18, 39] to irregular 3D point set analysis [29, 14, 51, 33, 30, 45]. Most of the methods [48, 25, 6, 28] transform the point cloud data to volumetric representations to implement weight sharing, leading to loss of information and high processing complexity. The local geometry of 3D object would be compromised due to quantization. Authors in [41] propose to transform point cloud data to grid representation. Multi-view CNNs [40, 4, 12, 46, 31] transform 3D point clouds to 2D images and apply 2D CNNs for point cloud analysis. However, by using these transformations we lose the underlying geometric information of the 3D point clouds, making it difficult to scale-up for scene understanding or point cloud analysis tasks.

Feature-based SVMs [9] for point cloud analysis, convert 3D point cloud data into a feature vector, namely metric tensors and christoffel symbols. And further use this feature vector for various 3D analysis using classical machine learning methods. We think that performance of this model would be constrained over the representation power of the features used.

Considerable amount of work has been done on point-based methods, i.e, to process 3D points directly with deep learning architectures. PointNet [27] proposed its first strategy towards this approach, where features were learned for each point followed by a symmetric function, giving a global signature for the whole point cloud. Yet, this strategy ignored the local context. Following this, PointNet++ [29] proposed a hierarchical feature learning network, extracting the local features and capturing fine geometric structure. SPFN [19] and ParSeNet [36] propose methodologies for primitive shape fitting to point clouds and [53, 44] learn to assemble objects by 3D volumetric primitives. SPFN [19] produces per-point segment membership and is trained to maximize IOU between predicted and ground truth memberships, however fitting primitive shapes to 3D objects makes to computationally costly and requires proper selection of parameters for each primitive shape, which makes it a difficult task to train a model. While ParSeNet [36] fits primitives with more robustness to point density and noise, it has limitations to fit boundaries smoothly and makes errors when parts are small requiring higher detail. To address these issues of 3D model decomposition task, our decomposition architecture directly predicts per-point primitive labels highly based on the local geometry of point clouds.

We propose Point Decomposition Network (PointDCCNet) for 3D object categorization. The key of PointDCCNet is decomposition of 3D object, into 4 basic primitive shapes and learning relations, i.e, local geometrical topology among these primitives, which in our view will encode each point, based on its neighbourhood points structure.

The key contributions of our work are as follows:

- We propose a 3D decomposition module which generates geometric signature of 3D objects using basic shapes namely plane, cone, sphere and cylinder. In order to build the decomposition module we do:
  - We extract 3D patches and generate patch features based on local neighbourhood and call this sub-module as Local Summarizer Layer (LSL).
  - We generate 3D point features by interpolating the acquired patch features by LSL and call this sub-module as Local Feature Propagation Layer (LFPL).
- We propose a 3D classification module which takes input features extracted from our proposed 3D decomposition module to categorize objects represented in point cloud format.
- We demonstrate the results of proposed 3D decomposition module using ANSI mechanical components dataset, and achieve an accuracy of 97.4%.
- We demonstrate the results of proposed 3D categorization architecture PointDCCNet using ModelNet40 dataset, and compare the results with other 3D object categorization techniques.

The rest of this paper is organized as follows. In Section 2, we provide the details of PointDCCNet architecture. Implementation details are discussed in Section 3. We discuss the results in Section 4 and conclude in Section 5.

2. Proposed PointDCCNet Architecture

The proposed PointDCCNet architecture for 3D object categorization is shown in Figure 2. PointDCCNet consists of 3 main blocks namely, 3D point cloud decomposition, feature encoder and classifier module for 3D object categorization. The concept of point cloud decomposition brings in an insight of certain low level geometric information. Our 3D point cloud decomposition module tries to extract features which are unique to a 3D object, represented as point clouds, helping the categorization task. In [10, 8, 35, 7] authors use decomposition framework to decompose a given point cloud into basic shapes (sphere, cone, cylinder) and use the decomposed primitives for different applications like 3D object super resolution [8], 3D inpainting of point clouds [7], 3D object categorization [10] and 3D object hole filling [35]. They propose to use a machine learning framework to decompose a 3D object using metric tensor and Christoffel symbols into a set of basis functions to perform the above mentioned tasks.

Consider a 3D object represented using point cloud. Let O be the set of point clouds of 3D objects, which contains

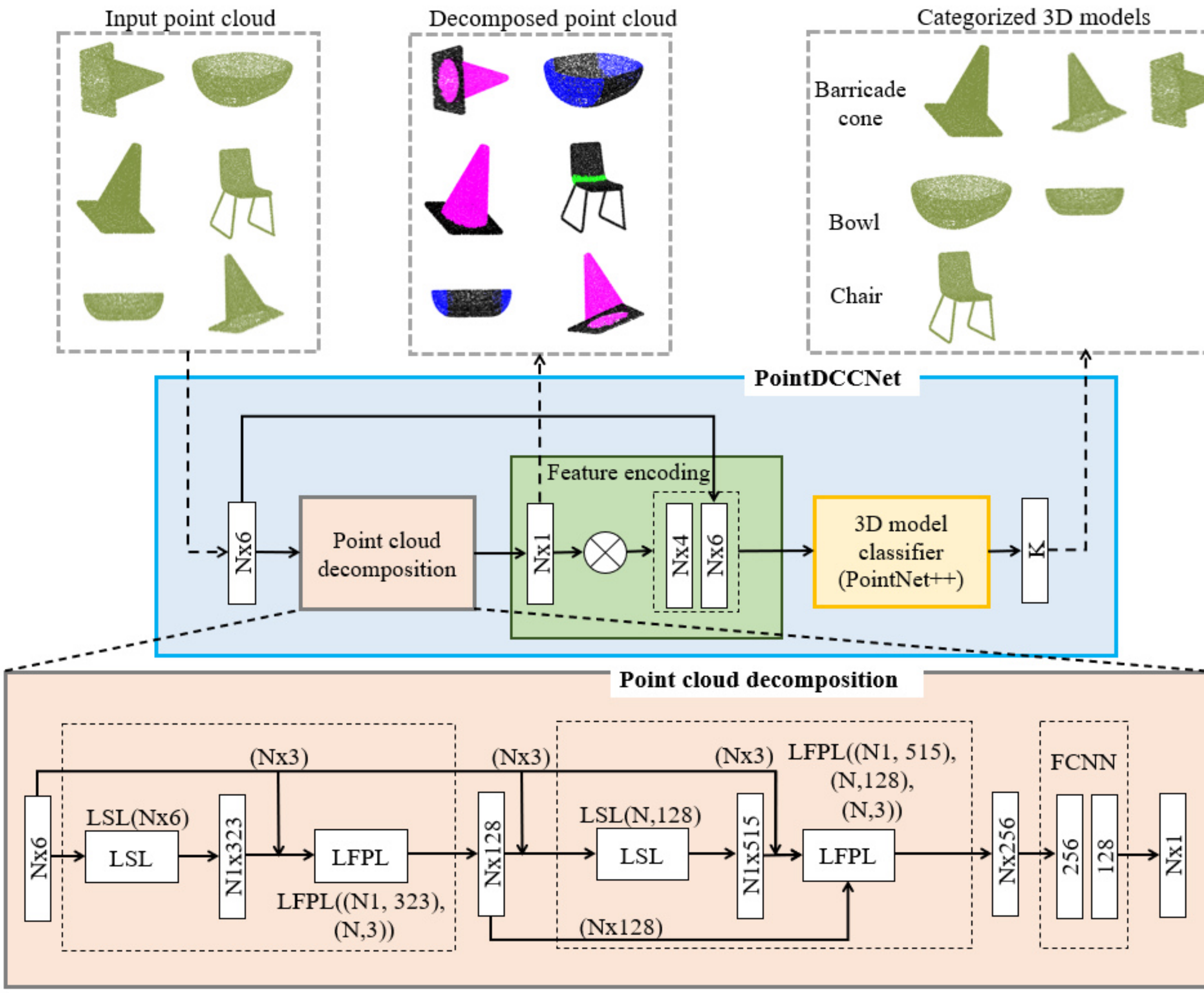

Figure 2. PointDCCNet architecture (point cloud decomposition as a plugin) for 3D object categorization. The symbol $\otimes$ represents one hot encoding function. For example 0 will be converted to [1, 0, 0, 0] vector, 1 will be converted to [0, 1, 0, 0] vector as similarly for others.

N number of points. Each point is characterized by 3 space points and 3 normals in x, y, z directions.

$$Pi \in O, 1 \leq Pi \leq N \tag{1}$$

Now, such Pi can be decomposed into primitive shapes (sphere, cone, cylinder) as represented in [8]. However, we choose to use planar primitive as another component because of [1], which says any 3D shape can be decomposed into four basic primitive shapes viz. plane, sphere, cone and cylinder

$$O = \{O1 \cup O2 \cup O3 \cup O4\} \tag{2}$$

where $O1$, $O2$, $O3$ and $O4$ are set of all planar, spherical, conical and cylindrical point clouds respectively.

To carry out the decomposition we propose a deep learning architecture shown in Figure 2 to decompose 3D object into primitive shapes. The properties for basic shapes like plane, sphere, cone and cylinder can be influenced from the local geometrical properties. The decomposition of the models is initially carried out on a local patch basis. The decision for the global classification of the model is derived from the results of the decomposition on the local basis. We let the deep architecture learn needed primitive representations of a given 3D object. This decomposition feature as a plugin with input point cloud and normals, will help the classifier to learn the inter-relations between each primitive shape, and will further boost classification performance.

We train the model to extract feature representation f for each point $p \in Pi$, where $Pi \in O^{N\times 6}$, where f encodes low level geometric information depending on neighbourhood of p. Using this feature representation, the model maps each point to its primitive shape label. These per-point primitive label information, provided by decomposition module is passed to feature encoding module, which transforms 3D object to a new latent space inheriting decomposition knowledge. This new latent representation is further used by 3D model classifier for better 3D object categorization. Our proposed modules are explained in detail below.

2.1. Point cloud decomposition module

The novel deep architecture for point cloud decomposition task as shown in Figure 2, takes in input point cloud P_i with point coordinates $N\times3$ and per-point normals $P_n \in \mathbb{R}^{N\times3}$, where N is number of points in P_i and predicts per-point primitive shape label $L \in \mathbb{R}^{N\times1}$. Our network supports $M = 4$ types of primitives: plane, sphere, cone and cylinder,

$$Pi \in O \implies \{P1 \cup P2 \cup P3 \cup P4\} \in O \quad (3)$$

where $P1, P2, P3, P4$ are the set of planar, spherical, conical and cylindrical points in given point cloud Pi.

The prediction of a point belonging to a primitive shape must be biased by its neighbouring geometric structure, which should be exploited by the model. PointNet++ [29] proposed a hierarchical learning method to learn per-point embedding which is influenced by local as well as global geometry. Employing this idea of hierarchical learning, we propose this novel decomposer module to extract the local shape-aware property of point clouds, constrained by smaller receptive fields. This ensure that the new learnt per-point embedding are influenced highly by local geometry.

To achieve this, the module has two consecutive layers of: Local Summarizer Layer (Section 2.1.1) and Local Feature Propagation Layer (Section 2.1.2).

2.1.1 Local Summarizer Layer (LSL)

In this layer we attempt to extract point features from the actual point cloud which are summarized from a local patch. The input point cloud $P_i \in O^{N\times d}$, with N points and 6 feature channels is summarized to a smaller set of points $P_s \in O^{N1\times d1}$, where $P_s < P_i$ and $d1 > d$. These new set of feature channels $d1$ are generated based on the neighbourhood points and represent point cloud in a higher dimension canonical space. This layer consists of 2 sub-layers: Sampling and Grouping layer and Pointnet layer.

Sampling and grouping layer. Given the input point set $P_i = \{x_1, x_2, \ldots, x_N\}$ we sample $N1$ points using the farthest point sampling algorithm to get a uniform subsampled point set $P_s = \{x_1, x_2, \ldots, x_{N1}\}$. Keeping these points as centroids, local clusters of size $N1 \times K \times d$ are formed using ball query, where K is number of points in each cluster. Ball query assigns all points to a centroid based on radius parameter r. Each cluster is then processed to make their respective centroids as origin and to get local-coordinate of the points with respect to the centroid. We use the novel MSG (Multiple Scale Grouping) layer [29] to capture patterns at different scales (varying radius). By setting a set of small grouping radius we force our network to capture local geometry and ensuring that the receptive field does not capture the whole point cloud.

Pointnet layer. All the $N1$ patches are further regressed individually using a set of MLP layers, followed by a maxpool function. We use this symmetric function proposed by Pointnet [27] to embed each patch to a common latent space. Input to this layer is of size $N1\times\mathrm{K}\times d$, which are the $N1$ local clusters to produce $N1\times d1$ sized point features. Features of all scales are concatenated to form a Local Summarized feature vector.

2.1.2 Local Feature Propagation Layer (LFPL)

After a first pass of point cloud from LSL the point set $P_i \in O^{N\times d}$ is reduced to $P_s \in O^{N1\times d1}$. However, for the task in hand we need to generate point features for all the original number of points. To get per-point features, we interpolate the summarized features to the original points $N\times d1$. Other methods to retrieve original number of points is to consider each point as centroid in LSL, but this increases the computational cost of system.

LFPL contains a set of fully connected layers that map the input point set to original number of points N. Skip connections are placed in order to traverse the information throughout the network. They propagate point coordinates and point features as shown in Figure 2.

Following to LPFL, the decomposer module has set of fully connected layers with dropouts to avoid overfitting. In the end, we get per-point primitive labels for point cloud.

2.2. Feature encoding module

In this module the input point cloud $N \times 3$ and its per-point primitive labels $N \times 1$ are processed. It uses one hot encoding technique, to transform per-point primitive labels to per-point primitive label vectors $N \times 4$. These per-point primitive label vectors are concatenated with the input point cloud $N \times 6$ giving a new latent space representation of the point cloud P_i of size $N \times 10$.

2.3. 3D model classifier module

Our classifier module is inspired by the design of Pointnet++ [29]. The input to this module are point coordinates $N \times 3$, normals $N \times 3$ and decomposition labels $N \times 4$ which is the new latent space representation given by feature encoding module. These decomposition feature as a plugin with input point cloud and normals, will help the classifier to learn the inter-relations between each primitive shape, and will further boost classification performance.

3. Experimental Details

In this section we discuss about the dataset used for training PointDCCNet architecture and also provide the implementation details of 3D point cloud decomposition module.

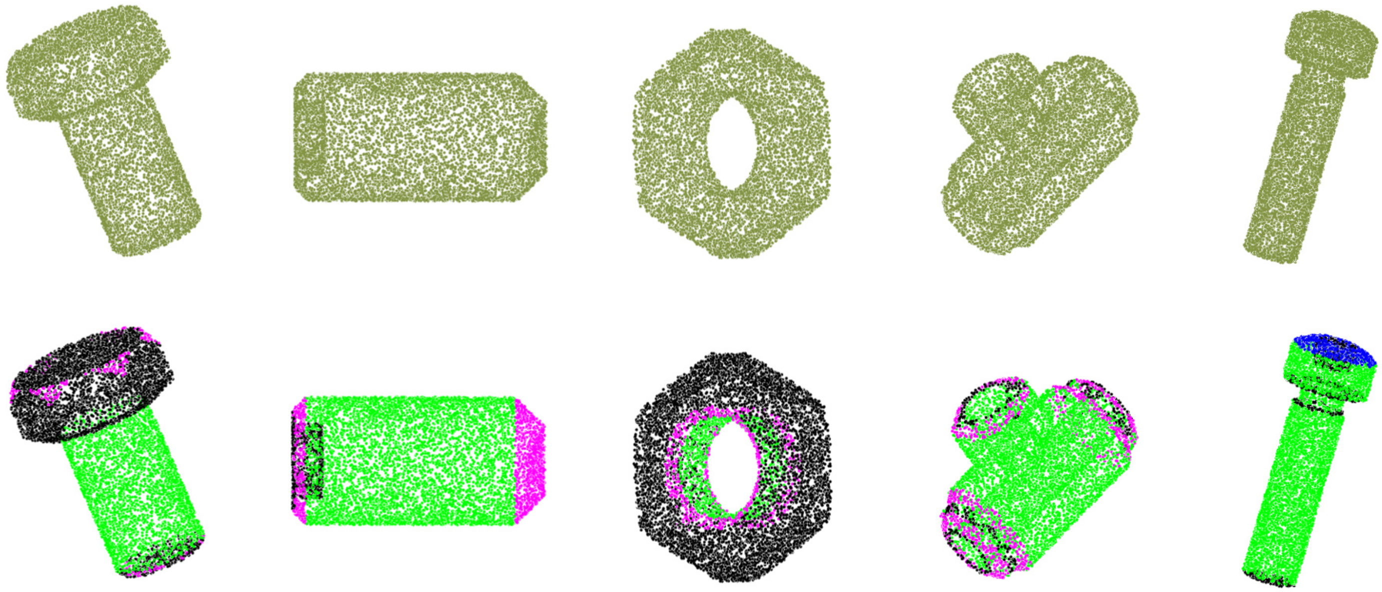

Figure 3. Results of decomposition using proposed point cloud decomposition architecture on ANSI mechanical components dataset. (Top row) Input point cloud (8096 points), (Bottom row) Decomposed point cloud [Green for Cylindrical, Magenta for Conical, Blue for Spherical and Black for Planar].

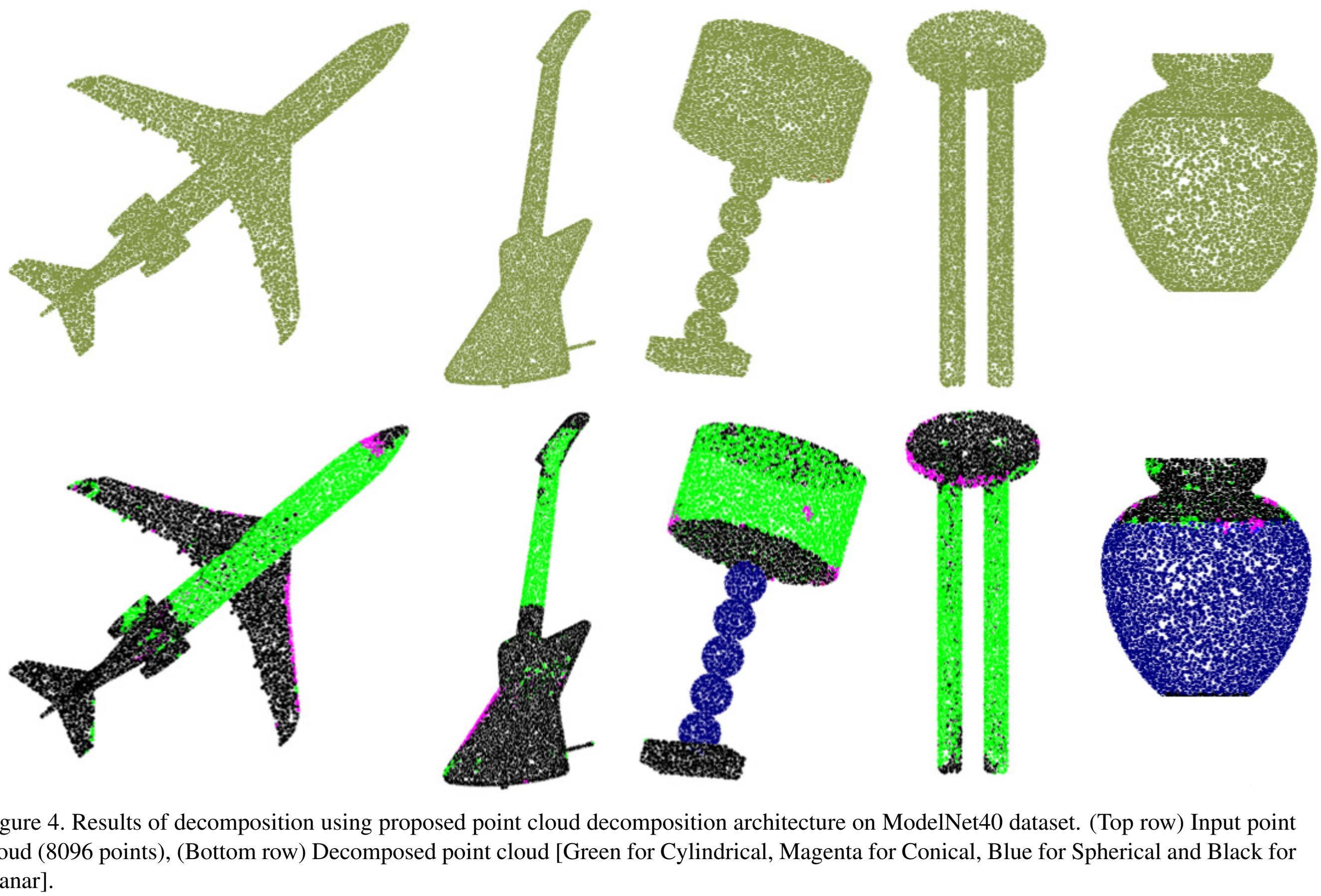

Figure 4. Results of decomposition using proposed point cloud decomposition architecture on ModelNet40 dataset. (Top row) Input point cloud (8096 points), (Bottom row) Decomposed point cloud [Green for Cylindrical, Magenta for Conical, Blue for Spherical and Black for Planar].

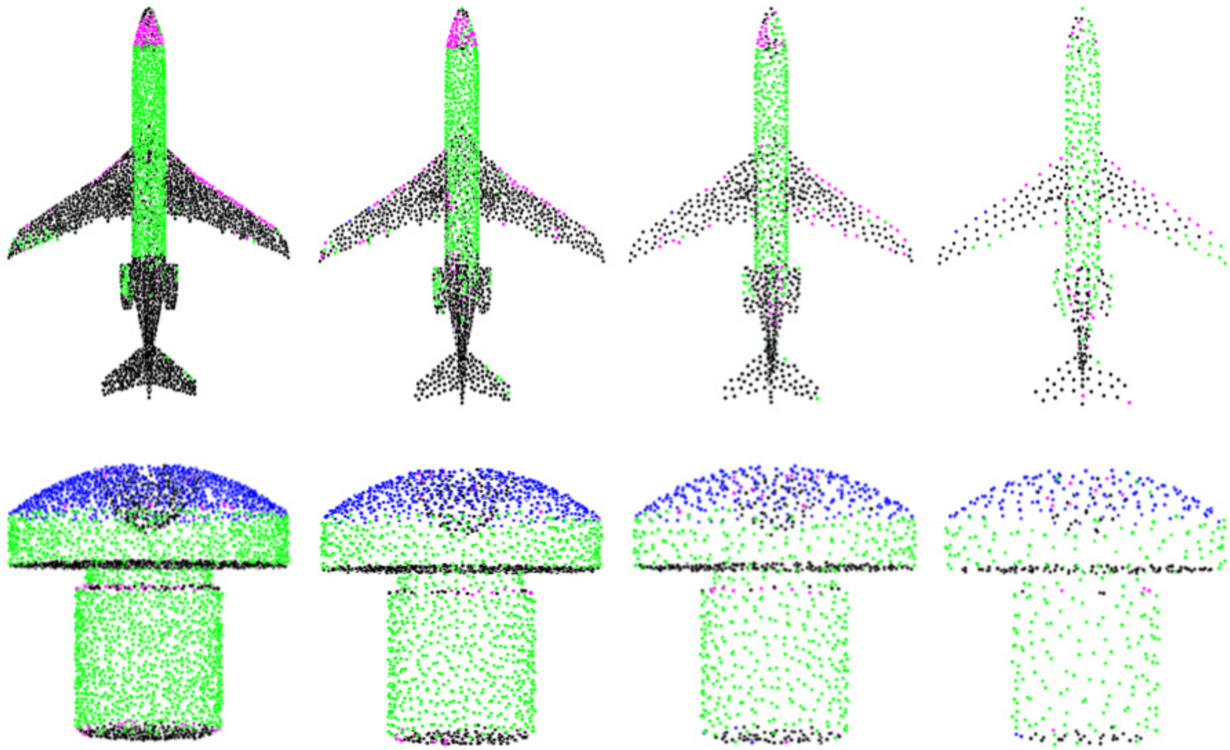

Figure 5. Results of point cloud decomposition using proposed architecture considering various point cloud densities, (left to right) 4096, 2048, 1024, 512. (Top row) ModelNet40 dataset and (Bottom row) ANSI mechanical components dataset.

3.1. Dataset

We choose to use American National Standards Institute mechanical component dataset, provided by Traceparts [43] to train our decomposer module. It includes 3D models of mechanical tools such as nuts, bolts shown in Figure 3, which ranges from smooth to steep rigid objects. We use a train/test split of 12984/3172 respectively. The categories are different in both sets, making training and testing sets disjoint. Each object has 8096 points, with their coordinates and normals. We have the associated ground truth primitive labels for training which is provided by Traceparts itself. As a part of pre processing we remove the models with more than 90% planar primitive category to prevent dataset skewness towards planar primitive.

For 3D model classifier module we utilize ModelNet40 [49] dataset, which consists of 12,311 CAD models with a total of 40 categories, where 9,843 objects are used for training and 2,468 for testing. We use point normals as additional feature along with point coordinates. From each 3D object we sample 1024 points as training inputs and train our variant of Pointnet++ for 3D object categorization.

3.2. Network and implementation details

Here we discuss the network and implementation details our decomposition module. This module has two sub-modules of which are LSL and LFPL. In the training phase of point cloud decomposition module, we set $N1 = 512$ for both the units, which are the number of centroids for local patches. We set $r = [0.05, 0.1, 0.15]$ and $r = [0.05, 0.1]$ for the first and second unit of LSL respectively for obtaining patch representations at different scales. In the Pointnet layer we use individual set of MLP's for each patch radius as follows: 1st unit MLP's $[32, 32, 64], [64, 64, 128], [64, 96, 128]$ and 2nd unit MLP's $[128, 128, 256], [128, 196, 256]$. Similarly there are 2 units of LFPL with set of MLP's which are: 1st unit MLP's $[256, 128]$ and 2nd unit MLP's $[512, 256]$. At the end, our PointDCCNet architecture has a FCNN unit (Fully Connected Neural Network) as another 3 MLP's which are $[256, 128, 4]$ has we have M = 4. We set a dropout of 0.4 in the last MLP layer.

During training both decomposer and classifier module, we augment the network inputs by random rotation, scaling and point perturbation with Gaussian noise. We train the decomposer module for 22 epochs using the Adam optimizer.

We set batch size to be 16 and learning rate to be 0.001. For the classifier module, we train the network for 100 epochs using Adam optimizer with a batch size of 24 and learning rate of 0.001. We implemented out network using PyTorch and trained it on Nvidia Quadro P5000 GPU.

4. Results and Discussions

In this section, we show the results of proposed PointDCCNet architecture using ModelNet40 and ANSI mechanical components dataset. We also compare the results with state-of-the-art techniques and show improved 3D object categorization.

4.1. Shape decomposition

We evaluate our decomposition model on ANSI mechanical component dataset as shown in Figure 3 and on ModelNet40 dataset as shown in Figure 4. Decomposition network achieves an accuracy of 97.4% on the Traceparts test set. In Figure 3, we can see that there is clear demarcation at the edges separating the primitive shapes, thus the decomposer is able to preserve sharp edges. However, it faces some difficulty when the surface complexity increases as seen in Figure 4. It is worth noting that in ModelNet40 dataset there is a smooth transition between primitive shapes, due to which the decomposer fails to map similar points into their respective primitives. Although the perplexity between spherical, cylindrical and conical is acceptable to some extent as they all have a positive curvature.

We test the robustness of the decomposition module on varying sampling density. With the increase in the sparsity of the point cloud, there is proportional increase in difficulty in surface prediction and thus, increasing the difficulty for shape decomposition. We observe that our proposed model is density-invariant and is shown in Figure 5; using sparse point clouds having total number of points 4096, 2048, 1024 and 512 respectively, as input to a model trained on 8096 points.

4.2. Shape classification

We evaluate our proposed PointDCCNet architecture for 3D object categorization task on ModelNet40 classification benchmark [48]. It consists of 9843 train objects and 2469 test objects having 40 classes. We uniformly sample 1024 points from each object and normalized to a unit sphere, for training and testing. The quantitative comparisons with the state-of-the-art techniques is shown in Table 1. We show improved results of object categorization over many techniques. As discussed in Section 2.3, we provide decomposition information to the classifier module. There is a significant increase in categorization performance by adding a decomposer network prior to the classification network. Our

Table 1. Results of 3D object categorization considering ModelNet40 benchmark dataset and comparison with state-of-the-art techniques (nor: normal)

Method	Input	#points	acc.
Pointwise-CNN [13]	xyz	1k	86.1
Deep Sets [52]	xyz	1k	87.1
ECC [38]	xyz	1k	87.4
PointNet [27]	xyz	1k	89.2
SCN [50]	xyz	1k	90.0
Flex-Conv [11]	xyz	1k	90.2
Kd-Net(depth=10) [17]	xyz	1k	90.6
PointNet++ [29]	xyz	1k	90.7
KCNet [37]	xyz	1k	91.0
MRTNet [6]	xyz	1k	91.2
Spec-GCN [45]	xyz	1k	91.5
PointCNN [20]	xyz	1k	91.7
DGCNN [47]	xyz	1k	92.2
PCNN [2]	xyz	1k	92.3
RSCNN [22]	**xyz**	**1k**	**93.6**
Spec-GCN [45]	xyz, nor	1k	91.8
Ours (PointDCCNet)	**xyz, nor**	**1k**	**92.5**
PointNet++ [29]	xyz, nor	5k	91.9
SpiderCNN [51]	xyz, nor	5k	92.4

PointDCCNet equipped with a shape decomposer achieves an accuracy of 92.5% in 3D object categorization. This shows that PointDCCNet classifier is able to exploit relation between shape primitives and object class distribution.

5. Conclusions

In this paper, we have proposed PointDCCNet for 3D object categorization empowered by its underlying geometric structure. This deep architecture learns the local topology of the object and provides this information to the classifier network. We have demonstrated the results of proposed PointDCCNet using benchmark datasets (ModelNet40 and ANSI mechanical components) and compared the results with state of the art techniques and show improved results for 3D object categorization. We have shown that, knowing the topological information shows improved performance for categorization task.

6. Acknowledgement

This project is partly carried out under Department of Science and Technology (DST) through ICPS programme - Indian Heritage in Digital Space for the project "CrowdSourcing" (DST/ ICPS/ IHDS/ 2018 (General)) and "Digital Poompuhar" (DST/ ICPS/ Digital Poompuhar/ 2017 (General)).

References

[1] William Abikoff. The uniformization theorem. *The American Mathematical Monthly*, 88(8):574–592, 1981. 3

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators, 2018. 7

[3] O. Chum and Jiri Matas. Matching with prosac - progressive sample consensus. volume 1, pages 220 – 226 vol. 1, 07 2005. 1

[4] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[6] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing, 2018. 2, 7

[7] Shankar Gangisetty and M. Uma. Example-based 3d inpainting of point clouds using metric tensor and christoffel symbols. *Machine Vision and Applications*, 29, 02 2018. 2

[8] Syed Altaf Ganihar, Shreyas Joshi, Shankar Setty, and Uma Mudenagudi. 3d object super resolution using metric tensor and christoffel symbols. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, ICVGIP '14, New York, NY, USA, 2014. Association for Computing Machinery. 2, 3

[9] Syed Altaf Ganihar, Shreyas Joshi, Shankar Setty, and Uma Mudenagudi. Metric tensor and christoffel symbols based 3d object categorization. In C. V. Jawahar and Shiguang Shan, editors, *Computer Vision - ACCV 2014 Workshops - Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part III*, volume 9010 of *Lecture Notes in Computer Science*, pages 138–151. Springer, 2014. 2

[10] Syed Altaf Ganihar, Shreyas Joshi, Shankar Shetty, and Uma Mudenagudi. Metric tensor and christoffel symbols based 3d object categorization. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, New York, NY, USA, 2014. Association for Computing Machinery. 2

[11] Fabian Groh, Patrick Wieschollek, and Hendrik P. A. Lensch. Flex-convolution (million-scale point-cloud learning beyond grid-worlds), 2020. 7

[12] Haiyun Guo, J. Wang, Yue Gao, J. Li, and H. Lu. Multiview 3d object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25:5526–5537, 2016. 2

[13] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks, 2018. 7

[14] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation, 2018. 2

[15] Adrien Kaiser, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. A survey of simple geometric primitives detection methods for captured 3d data. *Computer Graphics Forum*, 38(1):167–196, 2019. 1

[16] Zhizhong Kang and Zhen Li. Primitive fitting based on the efficient multibaysac algorithm. *PloS one*, 10:e0117341, 03 2015. 1

[17] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models, 2017. 7

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. 2

[19] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J. Guibas. Supervised fitting of geometric primitives to 3d point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 2

[20] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on $\mathcal{X}$-transformed points, 2018. 7

[21] Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4):to appear, 2011. 1

[22] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis, 2019. 7

[23] Anupama Mallik, Santanu Chaudhury, Vijay Chandru, and Sharada Srinivasan. *Digital Hampi: Preserving Indian Cultural Heritage*. Springer Publishing Company, Incorporated, 1st edition, 2018. 1

[24] Jiri Matas and O Chum. Randomized ransac with td,d test. *Image and Vision Computing*, 22:837–842, 09 2004. 1

[25] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[26] Luis E. Navarro-Serment, Christoph Mertz, and Martial Hebert. Pedestrian detection and tracking using three-dimensional ladar data. *The International Journal of Robotics Research*, 29(12):1516–1528, 2010. 1

[27] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. 2, 4, 7

[28] Charles R. Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data, 2016. 2

[29] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 2, 4, 7

[30] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds, 2017. 2

[31] R. B. Rusu, Nico Blodow, Zoltan-Csaba Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, 2008. 2

[32] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, 2009. 1

[33] Aleksandr Savchenkov, Andrew Davis, and Xuan Zhao. Generalized convolutional neural networks for point cloud data, 2018. 2

[34] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Comput. Graph. Forum*, 26:214–226, 06 2007. 1

[35] S. Setty, S. A. Ganihar, and U. Mudenagudi. Framework for 3d object hole filling. In *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1–4, 2015. 2

[36] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds, 2020. 2

[37] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling, 2018. 7

[38] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs, 2017. 7

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[40] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015. 2

[41] Rishabh Tigadoli, Ramesh Ashok Tabib, Adarsh Jamadandi, and Uma Mudenagudi. 3d-gcnn - 3d object classification using 3d grid convolutional neural networks. In Bhabesh Deka, Pradipta Maji, Sushmita Mitra, Dhruba Kumar Bhattacharyya, Prabin Kumar Bora, and Sankar Kumar Pal, editors, *Pattern Recognition and Machine Intelligence*, pages 269–276, Cham, 2019. Springer International Publishing. 2

[42] P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138 – 156, 2000. 1

[43] TraceParts S.A.S. Traceparts. 6

[44] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives, 2018. 2

[45] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning, 2018. 2, 7

[46] D. Wang and I. Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015. 2

[47] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 7

[48] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015. 2, 7

[49] Zhirong Wu, Shuran Song, A. Khosla, F. Yu, Linguang Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 6

[50] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7

[51] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters, 2018. 2, 7

[52] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018. 7

[53] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks, 2017. 2